第32卷 第16期 计 算 机 工 程 2006年8月
Vol.32 № 16 Computer Engineering August 2006
·安全技术· 文章编号：1000—3428(2006)16—0148—03 文献标识码：A 中图分类号：TP393.08

应用虚拟设备驱动的SSL VPN系统改进的实现

杨 杰，李 涛，王姝妲，王丽辉，杜 雨
（四川大学计算机学院，成都 610065）

摘 要：在分析传统SSL VPN不足的基础上，该文通过融合SSL VPN和IPSec VPN的设计思想，提出并实现了一种改进的SSL VPN系统。核心设计思想是采用虚拟设备驱动技术在Socket层进行数据包拦截，然后通过SSL加密隧道发送至隧道另一端，对方接收后进行解密和还原。与传统的SSL VPN系统相比，该系统具有更强的灵活性和适应性。改进的SSL VPN系统对于SSL VPN的发展和推广有着重要意义。

关键词：SSL；虚拟专用网；虚拟设备驱动

Improved Implementation of SSL VPN System Using Virtual Device Driver

YANG Jie, LI Tao, WANG Shuda, WANG Lihui, DU Yu
(Computer Department, Sichuan University, Chengdu 610065)

【Abstract】 This paper presents an improved SSL VPN system. The system hooks data packets from virtual device driver at socket layer and sends them though an encrypted SSL tunnel to the peer who will perform decryption and reconstruction of the packets. The system is more flexible and adaptable compared to traditional SSL VPN system.

【Key words】 SSL; Virtual private network(VPN); Virtual device driver

虚拟专用网[1](Virtual Private Networks, VPN)提供了一种通过公共非安全介质（如Internet）建立安全专用连接的技术，通过加密、认证、封装以及密钥交换技术在公众网上开辟一条隧道，使得合法的用户可以安全地访问企业的私有数据。IPSec VPN[1]和SSL VPN[2]作为两种广泛采用的VPN技术，各自有其优点，同时也有一定的不足。本文设计和实现了一种改进的SSL VPN系统。

1 传统SSL VPN的不足

通常情况下，当应用传统SSL VPN时，可以不用安装相应的客户软件，这样摈弃了安装软件的麻烦，但是也增加了对应用的支持不足。主要体现在传统的SSL VPN应用非常有限，仅适用于基于Web的应用，而一般的C/S程序是不能直接应用的。有的SSL VPN对Ftp，Telnet等服务进行了扩充，增强了其可用性。即便如此，也只能适用于数据库－应用服务器－Web服务器－浏览器这种模式。与IPSec VPN相比[3]，SSL VPN还存在网络扩展性不强的缺点，只能实现星型的PC－SSLGw－SERVERs[6]的应用模式，不适于搭建复杂多变的网络环境。

而以上SSL VPN的不足之处正好是IPSec VPN的优势所在。IPSec工作在网络层，对操作系统的所有网络封包进行处理，当然可以保护所有的网络应用，同时由于IPSec VPN本身的设计优势使得它非常适用于Lan－Lan的VPN模式，因此通过在SSL VPN系统中加入IPSec VPN的部分设计思想，使得这种改进的SSL VPN系统具有更强的可用性和适应性。

2 系统体系结构

整个系统由SSL VPN网关[4]和客户端软件构成。该系统可以在3种模式下工作[6]：

(1)Lan－Lan模式，这种模式下客户端不需要做任何安装和配置，仅需要对SSL VPN网关进行安装和配置。在两个SSL VPN网关之间，由SSL协议构建了一条安全通道，用来保护在局域网之间传送的数据。

(2)Web浏览器模式，此模式是SSL VPN的主要优势所在，由于Web浏览器的广泛部署，而且绝大部分Web浏览器内置了SSL协议，使得SSL VPN在这种模式下只要在SSL VPN网关上集中配置安全策略，几乎不用为客户端做什么配置就可使用，大大减少了管理的工作量，方便了用户的使用。

(3)客户端模式，此模式为远程访问提供安全保护，用户需要在客户端安装一个客户端软件，并做一些简单的配置即可使用，不需对系统做改动。这种模式的优点是保证所有建立在TCP/IP上的应用通信传输的安全。

图1描述了该系统的体系结构。

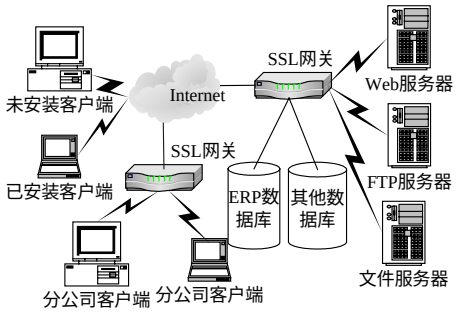

图1 系统体系结构

该系统特点是：各取所需——使用SSL VPN网关的企业

基金项目：国家自然科学基金资助项目（60373110）；教育部博士点基金资助项目（20030610003）
作者简介：杨 杰(1979—)，男，硕士生，主研方向：网络安全与人工智能；李 涛，博导、教授；王姝妲、王丽辉、杜 雨，硕士生
收稿日期：2005-09-10 **E-mail：**winrgn@163.com

分部可以和企业总部建立 Lan－Lan 的 VPN 连接，即 Lan－Lan 模式，这适用于分公司和企业总部之间的远程访问；未安装 VPN 客户端的用户可访问企业基于 Web 的资源，保持传统 SSL VPN 的便捷、高效、易于部署，即 Web 浏览器模式；安装 VPN 客户端的用户拥有更多的自由度，可以访问企业开放的所有资源，即客户端模式。 切换简单——未安装 VPN 客户端的用户可以通过软件安装轻松升级，不会对系统进行根本上的改动。统一策略——采用集中的安全策略管理，提高了安全性，降低了维护费用。

3 系统设计与实现

3.1 关键设计思想

基于Linux的IPSec实现的核心设计思想在于利用Linux提供的Netfilter框架[1]，在Netfilter框架提供的HOOK点上注册并实现IPSec相关处理函数。其基本工作原理是在网络层截获IP包，进行IPSec封装处理后再向物理链路中发送；接收数据包时同样在网络层截获IP包，经过IPSec还原处理后再传递给上层应用程序。本系统借鉴了这一设计思想，利用虚拟网卡驱动程序[5]，在Socket层使用虚拟网卡截获TCP/IP协议处理好的网络封包，交给SSL VPN处理模块，经过处理的封包再发送至物理链路或发送至上层应用程序。

3.2 SSL VPN 网关

要实现在Socket层进行截获数据包的操作，关键在于虚拟网卡的设计和实现。本系统的虚拟网卡驱动，数据接收和发送并不直接和物理网卡打交道，而是通过用户态来转交的。在Linux下实现核心态和用户态数据的交互，有多种方式[5]：

(1)通过 Socket 创建特殊套接字，利用套接字通信实现数据交互；

(2)通过 Proc 文件系统创建文件来进行数据交互；

(3)使用设备文件的方式，访问设备文件会调用设备驱动相应的例程，设备驱动本身就是核心态和用户态的一个接口。

本系统所使用的虚拟网卡驱动就是利用字符设备文件实现用户态和核心态的数据交互。

从结构上来说，虚拟网卡驱动并不单纯实现网卡驱动，同时它还实现了字符设备驱动部分。以字符设备的方式连接用户态和核心态。虚拟网卡驱动程序利用网卡驱动部分接收来自 TCP/IP 协议栈的网络数据包，发送或者反过来将接收到的网络封包传给 TCP/IP 协议栈处理；字符驱动部分则将网络数据包在内核态与用户态之间传送，模拟物理链路的数据接收和发送。

网络数据包的流程如下：在发送过程中，首先是使用虚拟网卡驱动的进程（即需要进行数据保护的网络通信进程）向虚拟网卡发送数据包，虚拟网卡并不会把 TCP/IP 协议栈处理好的数据包发送到物理链路，而是通过字符驱动模块把数据包发送至用户态，由 SSL VPN 处理模块进行加密封装等处理，最后再由该模块把处理好的数据包重新发送至 TCP/IP 协议栈，TCP/IP 协议栈处理后发送至物理网卡，从而进入物理链路。

接收数据包的过程与此类似，来自物理网卡的数据包通过 Socket 通信传递给 SSL VPN 处理模块，虚拟网卡驱动程序利用字符设备文件从 SSL VPN 处理模块获取还原处理后的数据包，最后再通过 TCP/IP 栈发送到使用虚拟网卡驱动的进程。该系统很好地实现了网卡驱动和字符设备驱动的结合。

图 2 为网络数据包流程示意图。

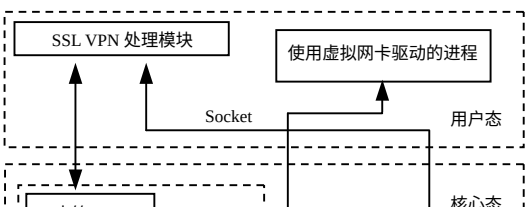

图 2 数据包流程

3.2.1 虚拟字符设备

在 Linux 中，字符设备和块设备统一以文件的方式访问，访问它们的接口是统一的，都是使用 open()函数打开设备文件或普通文件，用 read()和 write()函数实现读写文件等等。在本系统中，定义的虚拟字符设备的访问接口如下：

```
static struct file_operations tun_fops = {
    owner: THIS_MODULE,
    ioctl: tun_chr_ioctl,
    open: tun_chr_open,
    …
    read tun_chr_read,
    write: tun_chr_write,
    };
```

该结构中的每一个字段都对应驱动程序中实现特定操作的函数，如 tun_chr_read 函数对应读取设备文件的 read 函数，tun_chr_write 函数对应写设备文件的 write 函数等。利用 misc_register() 函数将该虚拟字符设备注册为非标准字符设备，提供字符设备具有的各种程序接口。

3.2.2 虚拟网卡

虚拟网卡驱动中实现的网卡驱动的主要处理例程：

```
tun_net_init ()；//网络设备初始化
tun_net_open ()；//打开虚拟网络设备
tun_net_close ()；//关闭虚拟网络设备
tun_net_xmit ()；//数据包发送
tun_net_status ()；//得到网络接口统计数据
```

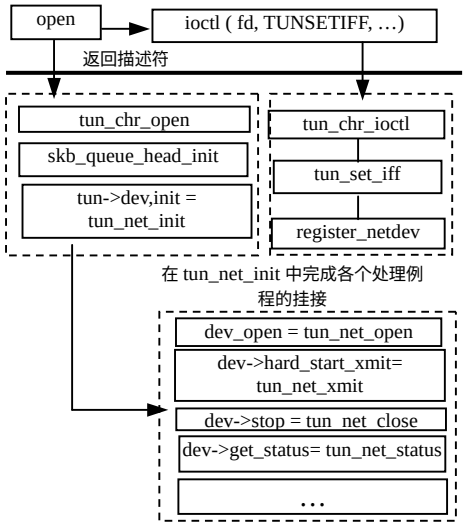

图 3 虚拟设备初始化

当打开一个虚拟设备时，open 函数将调用 tun_chr_open

函数，完成一些重要的初始化过程，包括设置网卡驱动部分的初始化函数、网络缓冲区链表的初始化、等待队列的初始化。虚拟网卡的注册被嵌入字符驱动的 ioctl 例程中，它通过对字符设备文件描述符利用自定义的 ioctl 设置 TUNSETIFF 标志完成网卡的注册。图 3 是函数调用关系的示意图。

使用 ioctl()函数操作字符设备文件描述符，将调用字符设备中 tun_chr_ioctl 来设置已使用 open 打开好的虚拟设备，如设置标志为 TUNSETIFF，则调用 tun_set_iff 函数，此函数将完成很重要的一步操作，就是对网卡驱动进行注册 register_netdev (&tun->dev)，此前，虚拟网卡驱动各个处理例程的挂接在 open 操作时由 tun_chr_open 函数初始化完成。

3.2.3 工作过程

从 TCP/IP 协议栈的角度而言，虚拟网卡驱动与真实网卡驱动并没有区别。从驱动程序的角度来说，它与真实网卡的不同表现在虚拟网卡获取的数据不是来自物理链路，而是来自用户区，虚拟网卡驱动通过字符设备文件来实现数据从用户区的获取。发送数据时虚拟网卡也不是发送到物理链路，而是通过字符设备发送至用户区，再由用户区程序通过其他渠道发送。工作过程如图 4。

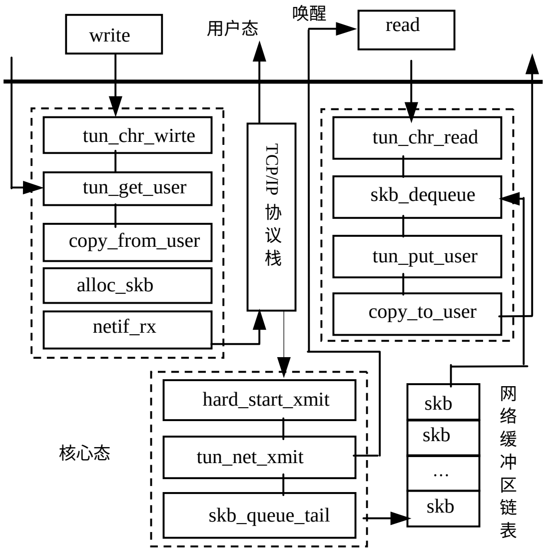

图 4 工作过程

(1)发送数据过程：使用虚拟网卡的进程经过TCP/IP协议栈把数据包传送给驱动程序，驱动程序调用注册好的 hard_start_xmit函数发送数据，hard_start_xmit函数又会调用 tun_net_xmit函数，其中数据包将会被加入skb（套接字缓冲区）链表[5]，然后唤醒被阻塞的使用虚拟字符设备驱动读数据的进程，接着虚拟字符设备驱动部分调用其tun_chr_read过程读取skb链表，并将每一个读到的数据包发往用户区，完成虚拟网卡的数据发送。

(2)接收数据的过程：当使用 write 系统调用向虚拟字符设备文件写入数据时，tun_chr_write 函数将被调用，它使用 tun_get_user 从用户区接受数据，并通过 alloc_skb 函数将数据存入 skb 链表中，然后调用关键的函数 netif_rx 将 skb 链表中的数据包发送给 TCP/IP 协议栈处理，最后由 TCP/IP 协议栈把数据包发送给使用虚拟网卡的进程，完成虚拟网卡的数据接收。

4 结束语

本文所设计并实现的 SSL VPN 系统不但保持了传统 SSL VPN 一贯的安全，便捷和易于部署的特点，同时克服了传统 SSL VPN 无法对所有应用层通信数据进行保护，以及不适用于构建复杂多变的网络环境等缺陷。基于 SSL 的 VPN 普遍被分析家看好，许多国际性大厂商纷纷迅速跟进，目前国外有一批厂商已经在提供基于 SSL 的 VPN 产品了。国内的 SSL VPN 发展目前尚处于起步状态，同时也受到很多具体情况的限制和约束，特别是 SSL VPN 本身的缺陷和不足。本文所提出的改进的 SSL VPN 系统对于 SSL VPN 的发展和推广有着积极意义。

参考文献

1 李 涛. 网络安全概论[M]. 北京: 电子工业出版社, 2004-08.

2 Andrew. SSL Virtual Private Networks[J]. Computers and Security, 2003, 22 (5): 416-420.

3 Jalal R. Net Security: IPSec vs. SSL[C]. Proceedings of the IEEE SoutheastCon-2004 "Engineering Connects", 2004: 351-358.

4 Ken A. SSL VPN Gateways: A New Approach to Secure Remote Access[J]. Database and Network, 2003, 33 (6): 3-5.

5 Rubini A, Corbet J. 魏永明, 骆 刚, 姜 君译. Linux 设备驱动程序[M]. 北京: 中国电力出版社, 2002-11.

6 包丽红, 李立亚. 基于 SSL 的 VPN 技术研究[J]. 网络安全技术与应用, 2004, 4(5)：42-47.

7 Cai Longzheng, Yu Shengsheng, Zhou Jingli. Research and Implementation of Remote Desktop Protocol Service over SSL VPN[C]. IEEE International Conference on SCC'04, 2004.

(上接第 138 页)

3 小结

属性证书作为兼具可扩展性和灵活性的访问控制方案的基础，它并不以用户或者群组分类区分访问条件，而是以属性集定义访问条件。属性证书综合考虑了访问控制系统中用户和其它的因素，给出了将访问授权信息传达到分布式应用的手段。这样，授权信息变得灵活机动而且具互操作性，比较适合于 Web Services 环境。

参考文献

1 柴晓路. Web Services架构与开放互操作技术[M]. 北京: 清华大学出版社, 2002.

2 Chadwick D W. An X.509 Role-based Privilege Management Infrastructure[Z]. Business BriefingGlobal Infosecurity, http://www.permis.org /, 2002.

3 Farrell S,Housley R. An Internet Attribute Certificate Profile for Authorization[EB/OL]. http://www.ieft.org/internet-drafts/draft-ietf-pkix-ac509prof-09.txt.

4 Jothy R, David L R. Securing Web Services with WS-Security[M]. Sams Publishing, 2004.